Count

我们不妨可以将每个 a_i 先对 m 取模，这样每个 a_i 的范围就是 $[1,m-1]$，我们令$A=\sum_{i=1}^{k} a_i\%m$，有 $A\%m=n\%m$，剩下 $(n-A)/m$ 个 m 可以随意分配给 k 个元素，这我们可以用组合数来计算。

现在的问题就变成了计算 a_i 在 $[1,m-1]$ 范围内，且 $A\%m=n\%m$ 的方案数，我们发现 A 的上限是 $k*(m-1)$，这个范围之内与 n 模 m 同余的只有 k 个数，我们不妨枚举这 k 个数，之后我们只需枚举有几个元素值大于 $m-1$ 来进行容斥即可。

时间复杂度 $O(k^2)$

Tree

由于要求所有路径的权值和，我们想到用点分治，到点分重心的路径我们可以用一个二元组 (A,B) 来记录，表示路径上的最大值和最小值，计算答案我们可以分四种情况分别计算，每种情况相当于二维数点，由于点权小于等于 5000，我们可以用二维树状数组来实现。

时间复杂度 $O(nlog^3n)$

Distance

题目要求所有点到两个关键点距离较小值的最大值，我们首先可以考虑一下哪些点离第一个关键点近，哪些点离第二个关键点近，我们发现我们可以找到两个关键点路径上的中点，中点的子树和其他点分成两部分，其中一部分离一个关键点近，另一部分离另一个关键点近。那问题就变成了这两部分的点到从属的关键点的距离的最大值。

我们定义:$fa[i][j]$ 为 i 这个点向上走 2^j 步到达的点，$up[i][j]$ 为从 i 出发，通过简单路径到达 $fa[i][0]$ 到 $fa[i][j]$ 这段路径上的点的子树中的点的最大值，$down[i][j]$ 为从 $fa[i][j]$ 出发，通过简单路径到达 $fa[i][0]$ 到 $fa[i][j]$ 这段路径上的点的子树中的点的最大值。

处理出这些数组之后，我们就能计算之前的问题了。

时间复杂度 $O((n+q)logn)$